Перекрестная ссылка на родственные заявки

Настоящая заявка является непредварительной заявкой, основанной на предварительной заявке № 60/631,152, поданной 26 ноября 2004 г., имеющей название «Способ создания таблицы данных с неподвижным заголовком в вебстранице», приоритет по которой настоящим испрашивается, а полное описание которой включено здесь путем ссылки.

Предпосылки создания изобретения

Начиная с создания сети Интернет и веб-страниц, существовала необходимость в отображении табличных данных в легко читаемой форме. При возникновении необходимости у разработчика веб-страницы отобразить несколько строк данных, например 20, 50, 100 или более строк исключительно желательно, чтобы заголовки, или метки каждого столбца строки данных оставались «зафиксированными», или «неподвижными», в то время как пользователь или лицо, использующее браузер, может прокручивать вниз строки данных, при этом имея возможность видеть и читать заголовки столбца или метки столбца в верхней части таблицы. Браузер может представлять собой любую из хорошо известных программ, таких как Microsoft Internet Explorer, Mozilla, Apple Safari, Opera, Firefox и т.п.

Ряд программ обеспечивает неподвижность строк заголовков в верхней части страницы при вертикальном прокручивании столбцов пользователем в таблице. В качестве одного такого примера можно привести Microsoft Excel, программу а на базе Windows, широко используемую в области бухгалтерского учета. Тем не менее, Microsoft Excel не является общепринятой веб-страницей, в которой используется HTML (гипертекстовый язык описания документов) или CSS (каскадные таблицы стилей), т.е. является прикладной программой, приемлемой для вывода данных на веб-странице Интернета. Несмотря на то что электронную таблицу Microsoft Excel можно все же рассматривать в браузере, тем не менее, она не является типовой веб-страницей, в которой используются HTML или CSS; она обладает многими ограничениями при ее использовании и характеризуется ограниченными областями применения.

Был предпринят ряд попыток создать веб-страницу, имеющую неподвижный заголовок. Некоторые разработчики использовали две таблицы HTML, при этом верхняя таблица содержала одну строку и обеспечивала установку параметров «ширины» каждого столбца и(или) его атрибутов таблицы до значения определенного пикселя или процента таким образом, чтобы ширина каждого столбца соответствовала значению ширины столбца нижней таблицы, содержащей данные. Недостаток указанного способа заключается в том, что пользователь сталкивается с проблемой определения точного значения ширины для установки параметров каждого столбца. Путем задания параметров ширины в статическом значении, либо в пикселях, либо в процентах, верхняя таблица и ее столбцы не в состоянии перемещаться, уменьшаться или увеличиваться по ширине с такой же скоростью, как если бы первоначально отсутствовали установке параметров ширины. Кроме того, при поступлении данных из базы данных также может иметь место постоянное изменение данных, следовательно, значения ширины столбцов верхней таблицы могут быть либо слишком большими, либо слишком малыми для обеспечения вывода данных и заголовков.

Другие способы вывода строки неподвижного заголовка таблицы включают способы, в которых используется HTML FRAMES и IFRAMEs. Существуют многочисленные веб-сайты, на которых содержатся перечни недостатков FRAMES при использовании веб-страниц. Ряд недостатков заключается в сложности при установке закладок и выводе на печать. Другие недостатки включают требования к «чистому» формату и дополнительным линейкам прокрутки. Дополнительный недостаток заключается в сложности разработки и сопровождении веб-страниц FRAMED или IFRAME.

Проблема еще более усложняется ввиду того, что существует острая необходимость в том, чтобы любой из способов вывода неподвижного заголовка обеспечивал бы работу при использовании в нескольких браузерах. Такая совместимость используемого способа с несколькими браузерами является исключительно труднодостижимой задачей даже без использования неподвижных заголовков. Проблема еще в большей степени усложняется тем, что каждый браузер имеет установку размера текста, которую пользователь может задать, и пользователь может игнорировать установки размера текста в браузере в зависимости от используемого браузера. Следовательно, даже если веб-программист, веб-разработчик и веб-дизайнер осуществит установку шрифтов для данных, отдельные пользователи могут игнорировать указанную установку в зависимости от используемого ими браузера с целью упрощения и ускорения процедуры просмотра веб-страницы при размере шрифта, при котором они желают ее просматривать.

Существуют многочисленные факторы, которые следует учесть при разработке таблицы с фиксированным заголовком в веб-странице. Ряд из указанных факторов приведен ниже.

1. Таблицы с фиксированными заголовками используются для вывода данных, которые обычно хранятся в базе данных определенного типа. Указанные данные могут постоянно изменяться. Следовательно, установление атрибута ширины <td> является исключительно сложным.

2. Также могут изменяться текст, размер шрифта и плотность шрифта (например, жирный) информационной строки заголовка. Это также негативно влияет на выравнивание столбцов.

3. Внутри ячейки данных могут быть также размещены изображения. Следует также учитывать этот фактор, т.к. в противоположном случае выравнивание будет нарушено.

4. Изображения могут быть размещены в заголовке (например, направление упорядочения). Если и в этом случае не учитывать этот фактор, произойдет смещение столбцов.

5. Различные версии браузеров и различные типы браузеров обеспечивают вывод данных в различном виде. Незначительные различия также могут привести к смещению столбцов.

6. Установки размера окна браузера пользователя отрицательно влияют на вывод данных. Пользователь, устанавливающий меньший или больший размер своего окна, может отрицательно повлиять на вывод данных. Создание максимального размера браузера также может привести к нарушению выравнивания столбцов.

7. Другие установки браузера, такие как установка размера текста, также могут повлиять на точное выравнивание столбцов. Установка размера текста задается пользователем, и она может, несомненно, повлиять на выравнивание столбцов. Реализация унифицированного описания шрифта HTML каждого браузера также является различной. То есть, определенные единичные размеры шрифта могут быть изменены некоторыми типами браузеров, в то время как другие браузеры не в состоянии изменить текст. Например, изменение IE Text Size не обеспечивает изменение размера шрифта, в котором используется единичный размер пикселя (px). Тем не менее, браузеры Mozilla и Opera позволяют своим пользователям изменять единичный размер шрифта пикселя.

Настоящий способ обладает следующими преимуществами:

1. Отсутствие необходимости использования PLUG-lns или запросов к пользователю загрузить программы, так как многие пользователи проявляют осторожность при загрузке неизвестных программ (например, вирусы, шпионящие ПО) в свои компьютеры.

2. Минимальная загрузка данных и (или) кодов на веб-страницу для более эффективного реагирования и (или) работы системы на веб-сервере и компьютере пользователя.

3. Минимальная нагрузка на центральный процессор веб-сервера с целью создания кода фиксированного заголовка, в результате чего обеспечивается обслуживание большего количества пользователей и (или) их более оперативное обслуживание.

Существуют такие программы, как PLUG-IN FLASH, которые обеспечивали реализацию таблицы с фиксированными заголовками. Но она не является стандартной HTML, и по этой причине пользователю требуется иной авторский инструментарий для того, чтобы приступить к работе с FLASH. В этом отношении, как известно, программа FLASH работает медленно. Более того, загрузка каких-либо программ из Интернета в компьютер является достаточно рискованной ввиду вирусов и шпионящих ПО. Такие программы, как ActiveX, Java Applets, обычно используются хакерами для инфицирования компьютеров пользователей, и ввиду такой ситуации пользователи осуществляют установку защиты, отвергающей такие программы, и приобретают коммерческое программное обеспечение защиты, препятствующее инсталлированию вышеуказанных программ на их компьютерах.

Известно, что Microsoft Excel можно просмотреть в браузере. Тем не менее, это является нежелательным, т.к. обеспечивается просмотр только страницы Excel. Кроме того, известно, что хакеры используют файлы Excel для передачи вирусов и шпионящих ПО. Следует отметить, что и язык Java может быть использован для этих целей, но опять же, это не простые HTML, CSS и JavaScript, которые приняты в качестве безопасных языков и уже установлены на компьютере пользователя.

Ряд программных обеспечений на веб-страницах предусматривает «двойной щелчок» мышью между столбцами заголовков с целью автоматического изменения размера столбцов для соответствия требованиям пользователя.

Указанный дополнительный шаг может оказаться неэффективным при определенных обстоятельствах, т.к. он усложняет относительно простую задачу. Веб-страницы, на которых предусмотрен такой шаг, в значительной степени зависят от JavaScript и практически не зависят от HTML. Такие страницы оказываются «тяжелыми», и загрузка в браузер пользователя занимает много времени. Несмотря на отсутствие подтверждения указанный шаг может не сработать на Macintosh, т.к. JavaScript, используемый в таких программах, является достаточно развитым и сложным. В результате этого, кросс-браузерная совместимость является трудно достижимой, т.к. JavaScript может оказывать различное воздействие на различные браузеры.

При создании таблицы с фиксированными заголовками (или сетки данных) необходимо учитывать указанные и иные факторы. Любой из указанных факторов, каким бы незначительным он ни был, может привести к смещению. И, таким образом, со времени появления веб-страниц до сих пор не был создан фиксированный заголовок, который может быть использован на уровне производственного качества. Таблица с фиксированным заголовком производственного качества должна найти применение в большинстве популярных браузеров (например, 99%) и обеспечивать подстройку к установкам многих пользователей, а также к выводимым данным.

Настоящее изобретение предусматривает устранение недостатков, связанных со способами предшествующего уровня техники, и создание способа отображения данных в таблице с первой горизонтальной остающейся неподвижной строкой, при этом позволяя пользователю прокручивать столбцы в вертикальном направлении. Любой пользователь в Интернете, использующий практически любой браузер, должен иметь возможности доступа к таблице с фиксированным заголовком, разработанной в соответствии со способом настоящего изобретения.

Краткое изложение существа изобретения

Таким образом, целью настоящего изобретения является создание способа отображения данных в табличной форме.

Другой целью настоящего изобретения является создание способа отображения табличных данных на веб-странице, при этом первая горизонтальная строка данных остается неподвижной.

Указанные и другие цели настоящего изобретения достигаются путем создания способа отображения табличных данных, при этом сохраняя заголовок в «неподвижном положении». В то же время обеспечивается прокручивание таблицы данных в вертикальном направлении при сохранении ее выравнивания с заголовком. Строка заголовка в верхней таблице может содержать несколько заголовков столбцов, соответствующих каждому столбцу данных ниже в соответствующей нижней таблице данных. В указанном способе таблица заголовков содержит, как минимум, две строки, и нижняя таблица данных содержит, как минимум, две строки. Одна строка верхней таблицы заголовков содержит и обеспечивает вывод просматриваемого текста заголовков. Вторая «скрытая» строка содержит текст указателя места заполнения (символ-заполнитель) или HTML для обеспечения автоматической установки ширины каждого соответствующего столбца заголовка. Текст указателя места заполнения или HTML каждого столбца заголовка «скрытой» строки определяется путем расчета максимальной ширины каждой ячейки данных и ее соответствующего текста или HTML в нижней таблице данных. Текст или HTML внутри ячейки, содержащей максимальную ширину для своего соответствующего столбца, далее используют в качестве указателя места заполнения в ячейке скрытых строк заголовков. В скрытой строке и в ячейке используются CSS, обеспечивающие создание ячейки, невидимой пользователем.

При наличии изображений, которые необходимо расположить в скрытой строке, высоту указанных изображений устанавливают на меньшее значение, например высота = "0" или высота = "1" или высота = "2" и т.д. Это приводит к уменьшению высоты скрытой строки при сохранении начальной ширины. Кроме того, при использовании другого способа CSS обеспечивается установка свойств видимости для скрытой строки или скрытой ячейки на значение «скрытый». Прокручиваемая часть таблицы данных создается с помощью DIV тегов и установки атрибутов CSS указанного DIV тега на значения, вызывающие появление линейки прокрутки. Атрибуты CSS обычно представляют собой высоту, переполнение, переполнение-х, переполнение-у и т.д. Такая информация может быть установлена с помощью программы или непосредственно в веб-странице.

Далее каждый текст столбца верхнего заголовка или ширину HTML сравнивают с шириной нижней таблицы данных по каждому столбцу. Если текст ячейки верхней таблицы заголовков или HTML шире, чем соответствующая ячейка нижней таблицы данных, к нижней таблице данных добавляют другую скрытую строку. Указанная «скрытая» строка для нижней таблицы данных содержит любой соответствующий HTML или текст в таблице заголовков с целью обеспечения изменения своего размера столбцами нижней таблицы данных для достижения ширины соответствующей ширине столбцов верхней таблицы заголовков. Две таблицы - верхняя таблица заголовков и нижняя таблица данных - обмениваются данными или метками заголовков и размещают их в их «скрытых» соответствующих столбцах и строках. В указанных скрытых строках используются свойства и значения CSS, такие как либо «видимость: скрытая», либо «вывод: отсутствует» для того, чтобы скрыть «переставленные» копии данных или меток заголовков. Более того, «переставленные» копии данных или меток заголовков не обязательно должны представлять собой точные копии данных или меток заголовков и могут являться любым HTML или текстом, которые обеспечивают соответствие ширины соответствующего столбца противоположной таблице, например верхней таблице заголовков или нижней таблице данных.

Настоящий способ также позволяет определить ширину текста и (или) HTML в каждой ячейке нижней таблицы данных. В частности, в тексте могут быть использованы различные типы пропорциональных шрифтов, например Anal, Times и т.д., не имеющие постоянной ширины, как, например, Courier. Способ позволяет рассчитать полную ширину всех символов в тексте для ячейки данных и использовать указанную полную ширину для определения того, какой HTML или текст ячейки должен быть скопирован и размещен в соответствующей ячейке противоположной таблицы.

Одним существенным компонентом способа является использование дополнительной строки «указателя места заполнения» данных, скрытых характеристиками каскадных таблиц стилей; вторым компонентом является расчетное использование наиболее широкой ячейки в столбце для размещения производного или скопированного содержания в смежной таблице в целях обеспечения надлежащего выравнивания столбцов данных и неподвижных заголовков.

В другом примере осуществления ширина каждого столбца таблицы (<TD> в HTML) в пикселях или процентах может быть установлена динамически в соответствии с расчетом ширины данных и ширины заголовков и путем выбора максимальной ширины из ячеек заголовков и ячеек данных. Если требуется установка в процентах, максимальная ширина ячейки каждого столбца могла бы быть суммирована, и далее процент мог бы быть определен для каждого заголовка и столбцов данных. Кроме того, максимальная ширина символа таблицы также может быть в дальнейшем модифицирована для укорочения исключительно протяженных данных через эллипс «...» или любого иного индикатора с целью уменьшения максимальной ширины столбца или столбцов.

При использовании данной системы таблиц неподвижных заголовков в соответствии с настоящим изобретением в сочетании с другой таблицей для третьей (3-й) таблицы, с определенными тегами CSS и DIV и с интеллектуальной обработкой также может быть создан многостолбцовый «выпадающий» список. Большинство, если практически не все выпадающие списки в Интернете, являются одностолбцовыми выпадающими списками. Имеется несколько списков, представляющих собой многостолбцовые выпадающие списки, тем не менее, существуют ограничения в отношении таких типов реализации. И как результат, в основном, отсутствуют широко распространенные или популярные веб-сайты, на которых используются такие многостолбцовые выпадающие списки.

Настоящий способ также обеспечивает вывод многостолбцовых выпадающих списков в веб-странице. Он обеспечивает выравнивание столбцов верхней строки данных со столбцами нижних строк данных, при этом дополнительно сокращая или расширяя нижние строки данных в зависимости от пользовательского входа. Способ позволяет пользователям видеть строки фиксированных заголовков в веб-странице, которые пользователи сейчас видят в программах электронных таблиц, таких как Excel, когда они вертикально прокручивают строки данных. За счет выравнивания столбцов строк фиксированных заголовков со столбцами нижних строк пользователи могут легко просмотреть названия столбцов, расположенные в строке фиксированных заголовков.

Дополнительная особенность настоящего изобретения позволяет пользователю выполнить расчеты изменения размеров на сервере и (или) на клиенте, при этом пользователю не придется выполнять ненужный двойной щелчок мышью с целью автоматического изменения размера столбцов или иные дополнительные мероприятия с целью получения ширины столбцов «правильного» размера.

Краткое описание чертежей

Ниже приведены ссылки на прилагаемые чертежи, на которых одинаковые детали обозначены одними и теми же позициями:

фиг. 1 - схематический вид отображения одностолбцовых данных, содержащего заголовок и один или несколько ячеек данных;

фиг. 2 - схематический вид многостолбцовой таблицы, при этом каждый столбец имеет заголовок и один или несколько ячеек данных;

фиг. 3 - схематический вид отображения одностолбцовых данных, имеющего скрытый заголовок и скрытую ячейку данных;

фиг. 4 - таблица фиксированных заголовков с видимыми скрытыми строками;

фиг. 5 - таблица со скрытым заголовком; существующая ячейка заголовков является отображением столбцов с максимальной шириной;

фиг. 6 - таблица с ячейкой данных, являющаяся максимально широким отображением в столбце;

фиг. 7 - таблица с содержанием скрытой ячейки данных, являющейся максимальным отображением столбца по ширине;

фиг. 8 - таблица, выведенная в коде Snippet каскадных таблиц стилей (CSS), используемых для скрывания скрытых строк;

фиг. 9 - схематический вид многостолбцового ниспадающего окна с таблицей фиксированных заголовков в сжатом и развернутом состоянии;

фиг. 10 - таблица HTML с различными столбцами в строках ячеек, имеющих максимальную ширину;

фиг. 11 - две таблицы, в которых в верхней таблице данные удалены, и нижней таблице удален заголовок;

фиг. 12 - таблица, иллюстрирующая пример ширины символов для шрифта "Geneva", приемлемого для использования в способе настоящего изобретения;

фиг. 13 - таблица, иллюстрирующая пример ширины символов для шрифта "Times", приемлемого для использования в способе настоящего изобретения;

фиг. 14 - образцовая расчетная таблица ширины вывода для фразы "New Orleans Saints;"

фиг. 15 - веб-страница, на которой пользователь имеет возможность изменять параметры установки размера текста, отображающие две таблиц выше, даже несмотря на то что они имеют одни и те же значения ширины <TD>, не обеспечивается их точное выравнивание;

фиг. 16 - таблица фиксированных врезок;

фиг. 17 - этап определения максимальной высоты ячейки в таблице основных данных;

фиг. 18 - этап процесса, в котором таблицу основных данных располагают сверху линейки прокрутки таблицы фиксированных врезок и столбца указателя места заполнения;

фиг. 19 - вид экрана компьютерного монитора с двумя таблицами фиксированных заголовков, при этом одна из таблиц выступает в качестве таблицы фиксированных врезок.

Подробное описание предпочтительных вариантов осуществления изобретения

Рассмотрим описание со ссылкой на чертежи более подробно. На фиг. 10 проиллюстрирована таблица HTML, имеющая заголовок 12, строки 14 и 16 и несколько вертикальных колонок столбцов A, B, C и D. Таблицы HTML естественно сокращаются вокруг наиболее широкой ячейки в столбце, если атрибут ширины столбца не установлен (или не описан) на определенную ширину пикселя или процент ширины.

Например, в столбце А ячейка в строке #2 является самой широкой. В столбце В ячейка в строке #1 является самой широкой. Тем не менее, в столбце С ячейка, содержащая текст "ColumnC" в строке заголовка, является самой широкой из всех строк, включая строку #1 и строку #2. В указанном примере таблица HTML без каких-либо описанных атрибутов ширины естественно сократится, или сожмется вокруг максимально широкой ячейки в отношении каждого столбца.

Настоящее изобретение предусматривает способ организации отображения, в котором единичная сетка данных создана с индивидуальными тегами и управляющими элементами для создания секции фиксированных заголовков. На фиг. 11 проиллюстрирована такая сетка данных, содержащая заголовок, и вторая сетка данных, расположенная непосредственно под сеткой данных, содержит заголовок с двумя строками и несколькими вертикальными столбцами. Ввиду того, что данные отделены от сетки данных, содержащей заголовок на этом рисунке, вполне естественно, что таблица уменьшается до минимальной ширины столбца при отсутствии описания других установок ширины таблицы и ячейки. Такие браузеры, как IE, Mozilla, Opera и т.д., позволяют отдельным пользователям устанавливать параметры размера текста содержания веб-страницы в соответствии со своими собственными установками. Это позволяет пользователям со слабым зрением адаптировать веб-страницы с исключительно малым размером шрифта к своим требованиям. Следует также учитывать указанные дополнительные установки параметров при создании таблицы фиксированных заголовков.

Способ отображения табличных данных в соответствии с настоящим изобретением может быть осуществлен на основе этапов «перестановки» и «скрывания». Во-первых, разработчик программного обеспечения или разработчик веб-страницы определяет максимальную ширину ячейки для каждого столбца в прокручиваемой таблице данных, которая превращается в видимую нижнюю таблицу. Фактически разработчик определяет, обмен какой информацией будет производиться между таблицами. Далее, копию содержания каждой ячейки, т.е. максимальная ширина для ее столбца, помещают в невидимую строку в таблице фиксированных заголовков, располагаемой над таблицей данных. Затем разработчик определяет, является ли содержание заголовка для каждой ячейки в таблице фиксированных заголовков (верхняя таблица) большей по ширине (текст или HTML), чем максимальная ширина ячейки прокручиваемой таблицы данных (нижняя таблица). Если содержание заголовка шире, то в этом случае разработчик размещает копию заголовка в невидимой строке в прокручиваемой таблице данных. Разработчик делает перекаченную информацию невидимой в строках holder-type (сборного типа), скрывающих переставленные копии в каждой таблице. В альтернативном случае разработчик корректирует переставленные изображения (если таковые имеются) с целью получения значения высоты в размере 1 (или в более общем плане меньшую высоту), учитывая, что определенные браузеры должны иметь высоту, как минимум, в размере единицы (1) с целью определения того, что в них находится изображение, требующее корректировки столбца. Ряд браузеров до сих пор реагирует на нулевую (0) высоту.

В результате «перестановки» имеется две таблицы, содержащие одинаковое количество столбцов. Если самая широкая ячейка в одном и том же столбце одной таблицы больше, чем в противоположной таблице, то в этом случае копию указанного текста размещают в противоположной таблице, увеличивая тем самым ширину противоположной таблицы, которая являлась бы шириной из исходной таблицы. Указанные шаги повторяют, за исключением из противоположной таблицы в исходную таблицу. Указанный процесс позволяет сопоставлять и выравнивать ширину текста или HTML для каждого столбца двух таблиц - таблицы заголовков и таблицы данных. В результате этого обеспечивается выравнивание ширины вертикальных столбцов обеих таблиц по отношению друг к другу. При необходимости порядок вышеприведенного алгоритма может быть изменен в программе, однако неизменными остаются способ и общая концепция.

В примере, приведенном на фиг. 1, табл. 20 образована с табл. 22 заголовков, и табл. данных 24 расположена под табл. заголовков 22. Табл. заголовков 22 содержит видимую ячейку А заголовков, обозначенную позицией 26, и скрытую ячейку А заголовков, обозначенную позицией 28. Нижняя табл. данных 24 содержит первую видимую ячейку А1 данных, обозначенную позицией 30, и вторую видимую ячейку А2 данных, обозначенную позицией 32. Скрытая ячейка А данных, обозначенная позицией 34, расположена под второй ячейкой данных 32. Скрытая ячейка данных 34 используется в качестве указателя места заполнения из данных, полученных из таблицы заголовков 22 и ячейки заголовков 26. Скрытая ячейка заголовков 28 увеличивается или уменьшается по ширине в зависимости от типа информации, размещенной в видимых ячейках данных 30 и 32. Обычно в качестве проводника выбирают более широкую из первой или второй ячеек данных 30, 32 и копию указанной информации размещают в скрытой ячейке заголовков 28. Аналогичным образом, если содержание видимой ячейки заголовков 28 шире любой из ячеек данных 30, 32, копию, или образец ячейки заголовков 26 размещают в скрытой ячейке данных 34.

В результате «перестановки» и «скрывания» разработчик создает две таблицы - верхнюю табл. заголовков 22 и нижнюю табл. данных 24 - имеющие идентичную ширину вывода и, соответственно, идентичную ширину по столбцам, в результате чего достигается выравнивание границ двух таблиц, а именно табл. 22 и 24.

На фиг. 2 проиллюстрировано выполнение вышеописанных шагов в многостолбцовой среде. В дан-

ном случае таблица заголовков 40 имеет три видимых ячейки заголовков A, B и C для соответствующих столбцов A, B и C, а также скрытые ячейки заголовков A, B и C для столбцов A, B и C. Таблица данных 42, расположенная под таблицей заголовков 40, имеет одну или несколько ячеек данных A1, B1, C1 и в альтернативных случаях ячейки данных A2, B2 и C2. Скрытые ячейки данных для каждого соответствующего столбца расположены под ячейками данных A2, B2 и C2. Для каждого столбца A, B и C определена максимальная ширина отображения столбца ячеек данных в таблице данных 42. Например, определяют максимальную ширину в тексте или HTML ячейки данных A1 или A2, и далее содержание скрытой ячейки A заголовков заполняют более широким значением ячейки данных AI или A2 с целью получения идентичной результирующей ширины, эквивалентной максимальной ширине столбца A таблицы данных.

На фиг. 3 проиллюстрирован пример, в котором производится копирование наиболее широкого отображения как из таблицы заголовков, так и из таблицы данных в скрытую строку/ячейку противоположной таблицы. Используя способ создания неподвижного заголовка, разработчик сначала определяет, имеет ли ячейка данных A1 в таблице данных 44 ширину отображения, превышающую ширину в ячейке данных A2. «Ширина отображения» является общей шириной символа и шириной изображения в строке. При наличии нескольких строк вбирают строку с максимальной общей шириной символа и изображений. Единицы ширины символа могут быть выражены в пикселях, точках, эмах или любой иной абсолютной или относительной единицы измерения. На фиг. 12-14 приведены примеры, иллюстрирующие размер каждой буквы и расчеты для фразы, которая может содержаться в данных или ячейке заголовков.

Если ячейка данных A2 шире в тексте или HTML, то в этом случае копию содержания, или образец содержания ячейки данных A2 размещают в скрытой ячейке A заголовков таблицы заголовков 46. В противоположном случае, если содержание ячейки A заголовков занимает большую ширину отображения, чем ячейка A1 данных или ячейка A2 данных, копию ячейки A заголовков или представление содержания ячейки A заголовков размещают в скрытой ячейке A данных под ячейкой A2 данных. Как в скрытой ячейке A заголовков, так и в скрытой ячейке A данных используют свойство каскадных таблиц стилей: видимость: скрытая; высота строки - 1 пиксель.

При наличии изображений, которые предусматривается разместить в скрытой ячейке A заголовков или скрытой ячейке B банных, производят изменение высоты изображений до высоты = "1" или высоты = "0". В отношении следующего столбца повторяют шаги сравнения ширины отображения и затем копируют содержание в скрытые ячейки заголовков или скрытые ячейки данных. В результате этого таблица заголовков 46 и таблица данных 44 имеют равную ширину и соответственно одинаковое выравнивание, ширину и выравнивание границ вертикальных столбцов.

На фиг. 4 приведен пример таблицы с фиксированным или неподвижным заголовком со скрытыми строками, видимым на экране пользователя. Область скрытой строки в строке 48 скрытого заголовка частично копируется из видимого заголовка и областей данных, т.е. областей, имеющих максимальную «ширину отображения». Первая видимая строка 50 содержит первый столбец 52, идентифицированный как "Order # ID", второй столбец 54, идентифицированный как «Customer ID», третий столбец 56, идентифицированный как "Ship Name", четвертый столбец 58, идентифицированный как "Order Date", и пятый столбец 60, идентифицированный как "Ship Country". Максимально широкий текст в третьем столбце 56 "Vins et alcools Chevalier", видимый в строке напротив Order # 10248 в видимой таблице 68 данных, регулирует ширину скрытой ячейки 62 заголовков и скрытой ячейки 64 данных в скрытой строке 66 данных.

В иллюстрации, приведенной на фиг. 5, не показана линейка прокрутки. На этом чертеже ячейка 70 заголовков является наиболее широким отображением соответствующего столбца 72. Ячейка 70 заголовков также шире, чем текст в таблице 74 данных для столбца 72. Скрытая ячейка 76 заголовков содержит данные, скопированные из ячейки 70 заголовков. Изображение было уменьшено до высоты "1", что позволит скрыть его в будущем. Этот же текст, являющийся максимально широким для столбца, воспроизведен в скрытой ячейке 78 данных, занимающей строку в самой нижней части столбца. Тем не менее, следует отметить, что скрытая ячейка данных может быть расположена в центре столбца или в самой верхней части столбца в зависимости от предпочтений разработчика.

На фиг. 6 представлена иллюстрация, на которой одна из ячеек данных является наиболее широкой в столбце. На этой иллюстрации ячейка 80 данных образует максимальную ширину отображения для столбца 82. Те же самые данные копируются в скрытой ячейке 86 заголовков и скрытой ячейке 88 данных. Несмотря на то, что ячейки 86, 88 скрытых таблиц заголовков и данных могут быть видны на фиг. 6, очевидно, что свойство CSS будет использовано для «скрывания» указанного содержания. Например, свойства CSS: видимость: скрытая; и высота строки: 1 пиксель могут быть использованы для того, чтобы скрыть скрытую ячейку 86 заголовков и скрытую ячейку 88 данных.

На фиг. 7 проиллюстрировано отображение, на котором одна из ячеек данных, в этом случае ячейка 90 данных, является самой широкой в столбце. Текст в скрытой ячейке 90 данных был получен из текста, появляющегося в определенном месте в столбце 92 и являющегося невидимым на иллюстрации, т.к. пользователь использовал линейку прокрутки 94 для перемещения вниз по столбцам 92 по отношению к тексту "Centro comercial Moctezuma". При наличии нескольких страниц данных в таблице максимально

широкое значение данных могло бы находиться в страницах, в настоящее время невидимых в браузере, даже если пользователь перемещает линейку прокрутки вниз. Производится корректировка скрытой ячейки 96 заголовков с целью достижения такой же ширины, как у скрытой ячейки 90 данных. Линейка прокрутки 94 может быть создана с помощью тегов <DIV> и атрибутов CSS: высота и переполнение-х и переполнение-у.

На фиг. 8 проиллюстрирован Code Snippet 98 каскадной таблицы стилей (CSS), который может быть использован для скрывания скрытых строк (строки указателя места заполнения). На этой иллюстрации скрытые строки ячеек заголовков и ячеек данных были скрыты и не видимы для пользователя.

На фиг. 9 проиллюстрировано многостолбцовое ниспадающее окно с неподвижным заголовком в сжатом и развернутом состоянии. Использование большего количества <DIV> тегов в многостолбцовом ниспадающем окне позволит скрыться или исчезнуть таблице данных и ее строке заголовков. В результате этого изображение на компьютерном терминале пользователя будет иметь вид сжатого состояния на фиг. 9. Дополнительный <DIV> тег окружает всю таблицу данных, а также таблицу заголовков. Другой программный способ или функция управляет атрибутами <DIV>, позволяя им оставаться скрытыми или не скрытыми.

На фиг. 12 и 13 приведен образец возможной ширины символа для шрифтов "Geneva" и "Times", обеспечивая единицы в пикселях для известных размеров шрифтов. Кроме того, различные шрифты имеют различную ширину. В рамках одного и того же шрифта кегли шрифта также могут иметь различную относительную ширину при сравнении с другими кеглями шрифта. Ввиду наличия исключительно большого количества различных кеглей и вариантов, предпочтительно сделать скрытую копию и позволить браузеру пользователя выполнить операцию установки кегля шрифта. На фиг. 14 проиллюстрирован примерный расчет ширины отображения для фразы "New Orleans Saints" при полном размере, составляющем 94,21223 пикселей.

На фиг. 12 и 13 также показано, что многие символы имеют одну и ту же ширину в рамках одной и той же группы шрифтов, или соответствуют ширине символов из других групп шрифтов. Соответственно, настоящее изобретение позволяет заменить исходные символы в скрытой ячейке 86 заголовков и скрытые ячейки 88 данных на другие символы или сочетание других символов с целью урегулирования ширины столбца. Также могут быть использованы зашифрованные или «представительные» данные для регулирования ширины столбцов. Следовательно, вместо копирования данных в скрытую ячейку 86 заголовков может быть проведено сопоставление между скрытой ячейкой 86 заголовков и максимальной ячейкой данных в столбце и далее «представительные» данные копируют в скрытую ячейку 86 заголовков. Аналогичным образом, аналогичное сопоставление может быть выполнено в отношении скрытой ячейки 88 данных и ячейки заголовков, в которых представительные данные копируются в скрытую ячейку 88 данных.

Настоящее изобретение также предусматривает создание альтернативного способа для формирования таблиц отображения данных, имеющих «неподвижные» заголовки. В соответствии с альтернативным способом программное обеспечение сначала определяет максимальную ширину ячейки для каждого столбца в прокручиваемой таблице данных, или нижней таблице. Далее программа определяет, является ли содержание заголовков для каждой ячейки в фиксированной верхней таблице больше по ширине, чем максимальная ширина ячейки в прокручиваемой таблице (нижняя таблица). При выполнении следующего шага программа рассчитывает минимальную ширину столбцов и производит установку каждого столбца в обеих таблицах на одну и ту же минимальную ширину столбца, тем самым обеспечивая выравнивание фиксированной верхней таблицы с прокручиваемой нижней таблицей.

Ввиду того, что установка ширины ячейки таблицы <td> также позволяет контролировать ширину столбца, программа вместо размещения данных указателя места заполнения в скрытой ячейке, как описывалось в первом примере осуществления настоящего изобретения, может просто рассчитать и определить ячейку с максимальной шириной каждого столбца, а затем рассчитать, каким должен быть атрибут ширины ячейки <td>, <ширина td="максимально рассчитанному количеству в пикселях">.

Предусматривается, что при альтернативном способе может потребоваться определенный вызов установки ширины ячейки таблицы, <ширина td=" в пикселях">. Например, если программист желает установить таблицу на 100% ширину, фиксированный <td>'s не должен перемещаться и растягиваться также легко и точно, как, прежде всего, при отсутствии установки ширины <td>. Такая ситуация может возникнуть, например, когда в ячейке содержится большой объем текста, который необходимо перенести на следующую находящуюся ниже строку.

В некоторых случаях желательно, чтобы на веб-сайте имелась таблица 100% ширины, которая занимала бы максимальное пространство по ширине. При этом в некоторых случаях желательно, чтобы таблица сократилась до минимальной ширины для экономии места.

Существуют многочисленные способы, с помощью которых программа может установить ширину столбца. Например, ширина столбца может быть определена путем установки атрибута ширины <td> на: (1) значение пикселя или (2) значение процента, либо (3) не устанавливая значений и позволяя содержанию внутри ячейки определить ширину ячейки. Кроме того, содержание смежных столбцов и значение атрибута ширины могут также отрицательно повлиять на ширину столбца, в частности на процентную

ширину. Даже впоследствии программист может столкнуться с ситуациями, в которых выводимые табличные данные имеют несколько строк, некоторые из которых являются короткими, некоторые из которых объединены с изображениями, при этом заголовки содержат изображения либо текст, либо сочетание изображений и текста. Рассчитанный размер шрифта каждого символа может изменяться в зависимости от браузера, размера окон, на который пользователь устанавливает в своем браузере, разрешающей способности и т.д. Каждый тип шрифта имеет едва заметные различия в зависимости от программы, включая превалирующие и стандартные совместимые браузерные программы. Указанные незначительные различия могут быть видны в программе или в браузере при отсутствии точного выравнивания столбцов заголовков со столбцами данных.

Таблицы, приведенные ниже, и иллюстрации на фиг. 15 имеют идентичные значения TD для атрибутов ширины. Код ячейки и значения ширины проиллюстрированы а таблице ниже:

```
<td width="50"> </td>
<td width="45"> </td>
<td width="65"> </td>
```


asf	12345	abcdefg

Как видно из двух таблиц выше, их столбцы выровнены друг с другом. В иллюстративных целях между двумя таблицами оставлено пространство для того, чтобы было видно, что они фактически разделены. Но на практике две таблицы расположены ближе друг к другу в вертикальном направлении и даже контактируют друг с другом. Тем не менее, существует возможность того, что при изменении пользователем установок размера текста в таблице выше с помощью опций браузера, приведенных на фиг. 15, две таблицы, приведенные выше, начнут расширяться с различной скоростью по мере того, как пользователь изменяет установки размера текста в своем браузере. Следовательно, столбцы двух вышеприведенных таблиц не будут выровнены. Если шрифты внутри фиксированного заголовка привязаны к установке пикселей в противоположность настраиваемой установке, предусматривается, что произойдет выравнивание столбцов таблиц, по меньшей мере, в браузере под названием "Internet Explorer" и, возможно, при использовании других браузеров. Таким образом, вместо установки параметров ширины каждого столбца программа должна определить максимальную ширину ячейки для каждого столбца, если программа предусматривает точное выравнивание всех столбцов.

Многостолбцовый «выпадающий» список, описанный в настоящей заявке, демонстрирует, что он может, в конечном счете, быть создан на уровне производственного качества, являющемся кросс-браузер совместимым. В многостолбцовом «выпадающем» вписке практически используются идентичные перестановка и синхронизация в отношении выбранных строк данных, как и те, что используются в отношении описанных выше в таблице неподвижных заголовков. Эта третья (3-я) таблица имеет скрытые строки через CSS, которые до сих пор заполнены точными данными, в результате чего обеспечивается выравнивание ширины столбцов с другими столбцами таблиц данных и столбцами таблиц заголовков. Кроме того, при выборе пользователем позиции в таблице данных указанная позиция заполнена в данной третьей (3-й) таблице.

Предусматривается, что для реализации настоящего способа создания таблиц данных с неподвижным заголовком могут быть использованы многие компьютерные языки. Следует отметить, что для реализации настоящего способа пользователь может использовать сервер с конфигурацией «клиент/сервер», клиентский браузер через такой язык, как JavaScript, либо Java, либо VBScript.

Кроме того, Furthermore, элементы указанного способа могут быть применены как на веб-сервере, так и на клиенте.

Имеется возможность выбора большого количества языков из серверных языков. Например, Java, J2EE, Perl, PHP, ColdFusion, Python, Visual Basic, ASP.NET, C#, J#, JSP, HTML, язык программирования или ассемблерный язык могут быть использованы для реализации описанного в настоящем документе способа. Данные для указанных таблиц обычно поступают с таких баз данных, как MS Access, SQL Server, IBM DB2, Oracle, MySQL и т.д. Тем не менее, они также могут поступить из XML файла, неструктурированного файла, JavaScript Object или из некоторых специальных форматов данных. Для поиска данных также может быть использован способ, называемый AJAX, асинхронный JavaScript и XML с целью повышения эффективности и улучшения времени реакции между сервером и браузером клиента.

Следует отметить, что между используемыми языками и базами данных существуют различия, тем не менее, может быть использован и используется один и тот же способ создания неподвижного заголовка и выравнивания столбцов.

Дополнительной особенностью настоящего изобретения является способ создания фиксированной врезки ячейки, подстраиваемой к высоте строки ячейки данных, но также прокручиваемой вниз и вверх и остающейся в вертикальной синхронизации с основной таблицей фиксированных заголовков, в то время как основная таблица заголовков прокручивается влево или вправо для просмотра скрытых столбцов. Эта задача может показаться несколько более простой, чем создание фиксированного заголовка, т.к. обычно может быть только 1, 2, 3 и т.д. строк по высоте. Тем не менее, при реализации полной программы электронных таблиц, такой как Excel, обладающей большим набором характеристик с помощью настоящего изобретения, программа могла бы быть в значительной степени усовершенствована.

При формировании фиксированной врезки используют ту же самую концепцию перестановки для столбца указателя заполнения места. Программа отыскивает номер (число) разрывов (переносов) строк и, соответственно, выполняет перестановку ячеек данных. Следовательно, существуют две примыкающие таблицы фиксированных заголовков, в которых, например, вся левая таблица фиксированных заголовков также функционирует в качестве фиксированной врезки. Линейка прокрутки для левой таблицы заголовков либо удалена, либо скрыта, однако с помощью определенного языка программирования положение линейки прокрутки таблицы фиксированной врезки остается синхронизированным с положением линеек прокрутки основной таблицы фиксированных заголовков.

На фиг. 16 и 17 показана табл. 100 фиксированных врезок, включающая столбец 102 символов-заполнителей и столбец 108 фиксированных врезок. Столбец 102 символов-заполнителей имеет заголовок "Customer ID" и несколько ячеек 104 данных. Одна из ячеек 105 данных имеет максимальный объем информации и, следовательно, имеет максимальное значение высоты.

Перемещаемая линейка прокрутки 106 показана справа от столбца 102 символа-заполнителя. Столбец 108 фиксированной врезки показан слева от столбца 100 фиксированной врезки. Основная таблица 110 данных может содержать дополнительные столбцы, как показано на фиг. 17, которая дополнена столбцом 112 "Ship Name", столбцом 114 "Order Date" и столбцом 116 "Ship Country". Несомненно, основная таблица может содержать только один столбец в зависимости от характера отображаемых данных.

В процессе разработки фиксированной врезки один из шагов предусматривает скрывание столбца 102 символа-заполнителя и линейки прокрутки 106 табл. фиксированных врезок основной табл. 110 данных с помощью свойств CSS, например слоев z-индекса CSS, при этом сохраняя видимым столбец 108 фиксированных врезок. Столбец 102 символа-заполнителя создают путем определения ячейки в основной табл. 110 данных или табл. 100 врезок, имеющей максимальное значение высоты. На этой иллюстрации ячейкой является ячейка под позицией 105. Аналогичным образом, если ячейка содержит только текст, значение определяется на основе количества строк возвратов каретки или HTML <BP> или <P> тегов.

Как видно из фиг. 18, основная табл. 110 данных частично закрывает (расположена «над») табл. 100 фиксированных врезок, т.е. столбец 102 символа-заполнителя и линейку прокрутки 106 табл. 100 фиксированных врезок. Следовательно, когда элементы основной таблицы закрывают линейку прокрутки 106 табл. 100 фиксированных врезок, табл. 100 и табл. 110 могут появиться в виде одной таблицы.

На фиг. 19 проиллюстрирован другой пример осуществления настоящего изобретения для отображения страницы, на которой представлены две таблицы фиксированных заголовков, при этом одна из таблиц выступает в качестве таблицы фиксированных врезок. На этом рисунке большая табл. 112 имеет табл. 114 фиксированных врезок и несколько столбцов 115 данных. Табл. 114 фиксированных врезок выступает в качестве фиксированного столбца, который не изменяется при использовании горизонтальной линейки прокрутки 116 или при перемещении влево или вправо. Горизонтальная линейка прокрутки 116 расположена в нижней части таблицы, позволяя пользователю перемещать в горизонтальном направлении влево или вправо основную табл. 118 данных с целью просмотра других столбцов данных. Вертикальная линейка прокрутки 120 позволяет пользователю просматривать основную таблицу, перемещая ее вверх или вниз, в то время как заголовок может оставаться зафиксированным или не зафиксированным в зависимости от выбора разработчика программ.

Настоящее изобретение позволяет разработчику программ создать врезку столбца с двумя таблицами фиксированных заголовков, в которых используются идентичные данные, но, как показано, используются две примыкающие таблицы. Основная таблица «перекрывает», или закрывает таблицу фиксированных врезок, в результате чего скрыты столбец символа заполнителя и линейка прокрутки таблицы фиксированных врезок с помощью ряда свойств CSS (или иных аналогичных средств), которые могут быть использованы для скрывания столбца символа-заполнителя и линейки прокрутки таблицы фиксированных врезок.

Кроме того, вертикальная линейка прокрутки основной таблицы синхронизирована с линейкой прокрутки фиксированной врезки, что обеспечивает выравнивание строк в обеих таблицах по отношению друг к другу. Это может быть выполнено с помощью способа перестановки данных и поиска ячейки

с максимальной высотой и путем введения данных в опцию таблицы фиксированных врезок, а также скрытого 2-го столбца. Обе таблицы имеют вертикальное положение линейки прокрутки по отношению друг к другу при прокрутке вниз или вверх. Это может быть осуществлено посредством клиентского JavaScript script в веб-странице с использованием разнообразных способов.

Путем горизонтальной прокрутки влево или вправо в основной таблице данных обеспечивается перестановка заголовка и данных таблицы фиксированных заголовков из скрытого столбца в следующее положение видимого столбца, в то время как производится скрывание видимого столбца в основной таблице данных (влево или вправо в зависимости от направления прокрутки).

Могут быть внесены различные изменения и модификации в способ настоящего изобретения в пределах его существа. Таким образом, права на настоящее изобретение будут ограничены только объемом прилагаемой формулы.

ФОРМУЛА ИЗОБРЕТЕНИЯ

1. Способ отображения скомпонованных в таблицу записей на управляемом компьютером экране, при этом таблица содержит столбцы, строки и несколько ячеек, образуемых столбцами и строками, при этом способ включает следующие шаги:

формирование первой непрокручиваемой подтаблицы для отображения заголовков столбцов;

формирование второй прокручиваемой подтаблицы для отображения записей, соответствующих заголовкам первой подтаблицы;

определение параметров максимального размера для каждого столбца в первой подтаблице и второй подтаблице на основе параметров максимальной ширины или высоты ячейки внутри каждого столбца, тем самым создавая ячейку максимального параметра;

определение видимой части первой подтаблицы, которая видима на управляемом компьютером экране, и скрытой части первой подтаблицы, которая невидима на управляемом компьютером экране;

копирование данных ячейки максимального параметра в ячейку внутри соответствующего столбца в скрытой части первой подтаблицы, определяя тем самым параметры ширины или высоты соответствующего столбца, и

отображение на управляемом компьютером экране записей, скомпонованных в таблицу, имеющую непрокручиваемую часть и прокручиваемую часть в каждой ячейке.

2. Способ по п.1, в котором данным параметрам, копируемым в скрытую первую подтаблицу, заданы предварительно определенная высота, скорректированная по отношению к предварительно определенному значению атрибута высоты или ширины до копирования данных параметров в скрытую часть первой подтаблицы.

3. Способ по п.2, в котором значение атрибута высоты или ширины имеет значение высоты в размере 1 пикселя или менее.

4. Способ по п.1, в котором ширину ячейки задают путем установки атрибута ширины <td> до значения пикселя записи внутри ячейки.

5. Способ по п.1, в котором ячейке задается дискретная ширина путем определения процентного значения соответствующего столбца по отношению к остальной ширине таблицы данных.

6. Способ по п.1, в котором непрокручиваемая первая подтаблица определяет фиксированный заголовок таблицы.

7. Способ по п.1, в котором первая подтаблица определяет фиксированную врезку таблицы.

8. Читаемая компьютером среда, включающая компьютерную программу, способную определять наборы параметров высоты или ширины для записи базы данных, расположенной в таблице данных, имеющей столбцы, строки и несколько ячеек, образованных столбцами и строками для определения максимальных параметров высоты или ширины для каждого столбца или строки в таблице данных на основе записей, отображаемых в ячейке, имеющей максимальные параметры высоты или ширины для создания первой подтаблицы, образующей непрокручиваемую часть каждого столбца; для создания второй подтаблицы, образующей прокручиваемую часть каждого столбца; для создания видимой части первой подтаблицы и скрытой части первой подтаблицы; и для копирования записей из ячейки, имеющей данные максимальных параметров ширины или высоты в ячейку внутри соответствующего столбца в скрытой части первой подтаблицы с целью определения параметров высоты или ширины соответствующих столбцов или строк в первой подтаблице и второй подтаблице, при этом указанная читаемая компьютером среда определяет набор параметров для отображения записей, скомпонованных в таблицу, имеющую неподвижную часть и прокручиваемую часть в каждом столбце.

9. Читаемая компьютером среда по п.8, в которой записям, копируемым в скрытую первую подтаблицу, задана предварительно определенная высота, скорректированная по отношению к предварительно определенному значению атрибута, до копирования записей в скрытую часть первой подтаблицы.

10. Читаемая компьютером среда по п.9, в которой значение атрибута имеет значение высоты в размере 1 пикселя или менее.

11. Читаемая компьютером среда по п.8, в которой ширина ячейки определена путем установки ат-

рибута ширины <td> до значения пикселя записи внутри ячейки.

12. Читаемая компьютером среда по п.8, в которой ширине ячейки задают дискретную ширину, которая определена путем определения процентного значения соответствующего столбца по отношению к остальной ширине таблицы данных.

13. Читаемая компьютером среда по п.8, в которой непрокручиваемая первая подтаблица образует фиксированный заголовок таблицы.

14. Читаемая компьютером среда по п.8, в которой непрокручиваемая первая подтаблица образует фиксированную врезку таблицы.

15. Читаемая компьютером среда по п.8, в которой записи, копируемые в скрытую первую подтаблицу, копируют в скрытую часть первой подтаблицы.

16. Способ по п.1, в котором данные ячейки максимального параметра, копируемые в ячейку внутри соответствующего столбца в скрытой части первой подтаблицы, являются репрезентативными данными ячейки максимального параметра.

17. Способ создания строк неподвижных заголовков в веб-странице для таблицы данных, имеющей несколько ячеек данных, при этом каждая имеет прокручиваемые вертикальные столбцы и горизонтальные строки, при этом способ включает

генерирование, как минимум, одного символа заполнителя в создаваемой ячейке таблице данных;

копирование в строку неподвижного заголовка, как минимум, одного символа-заполнителя из таблицы данных;

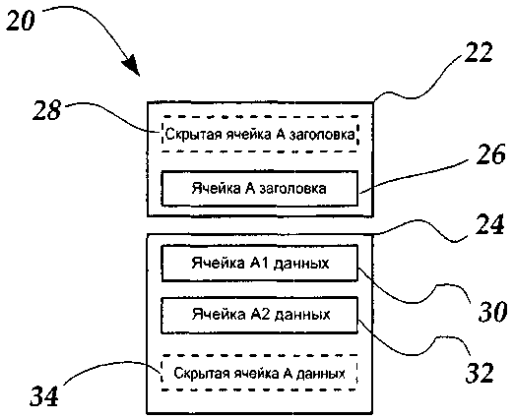
введение, как минимум, одного символа-заполнителя в соответствующий столбец таблицы с целью автоматического установления столбцом значения ширины, идентичного соответствующим создаваемым ячейке таблицы и столбцу;

расчет максимальной ширины текста для каждой ячейки данных и

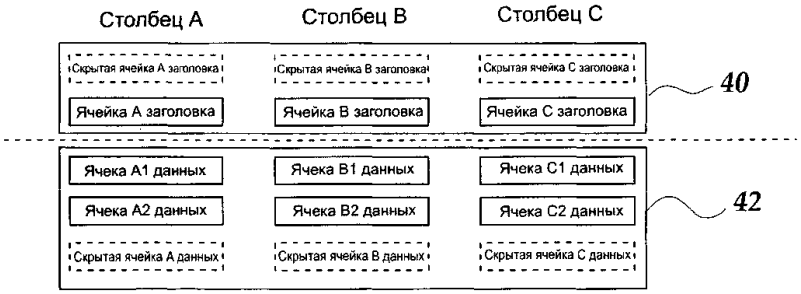
создание неподвижной части и прокручиваемой части в каждой ячейке данных.

18. Способ по п.17, в котором символ-заполнитель генерируют в строке стационарного заголовка и копируют в таблицу данных с целью сбалансирования и выравнивания ширины ячеек для каждого столбца в неподвижной части и прокручиваемой части и с целью достижения выравнивания между неподвижной частью и прокручиваемой частью.

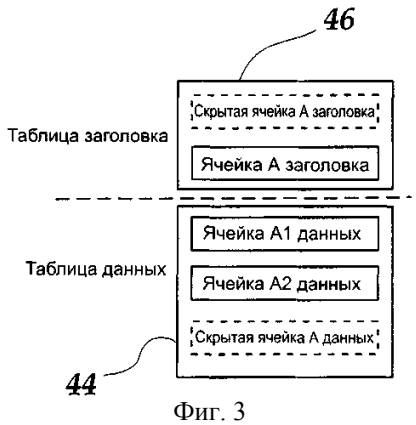
19. Способ по п.1, в котором данные ячейки максимального параметра из видимой части первой подтаблицы копируют в скрытую часть второй подтаблицы с целью сбалансирования и выравнивания ширины ячеек для каждого столбца в неподвижной части и прокручиваемой части и с целью создания выравнивания между скрытой частью и видимой частью таблицы.



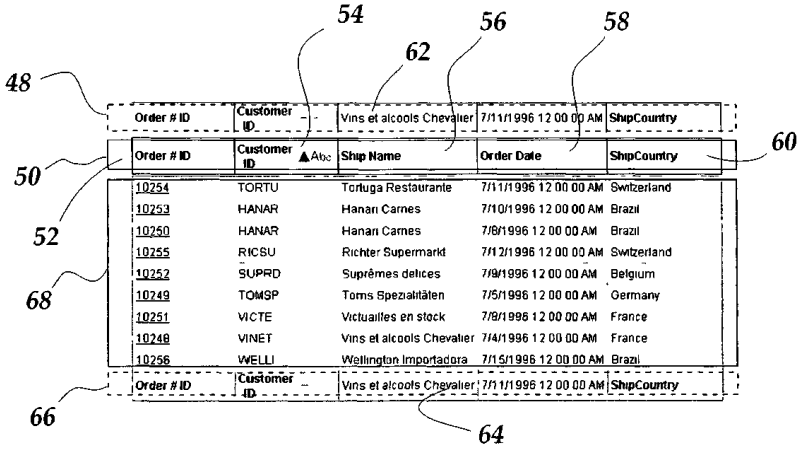
Фиг. 1



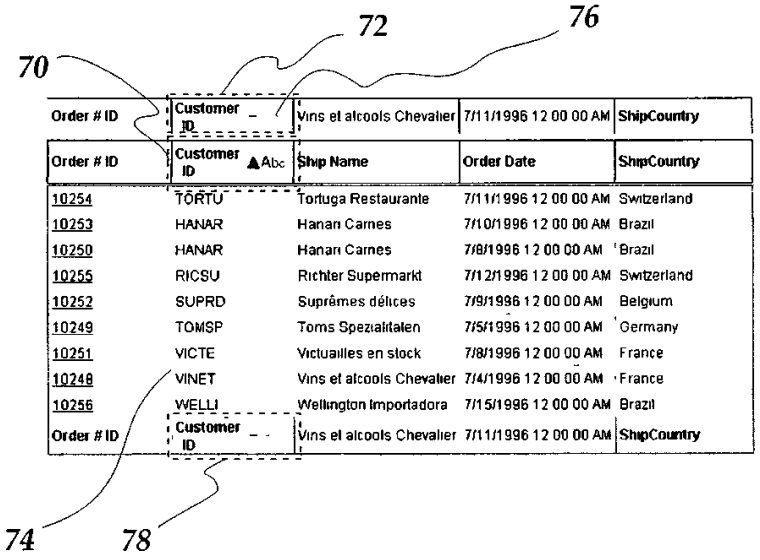
Фиг. 2



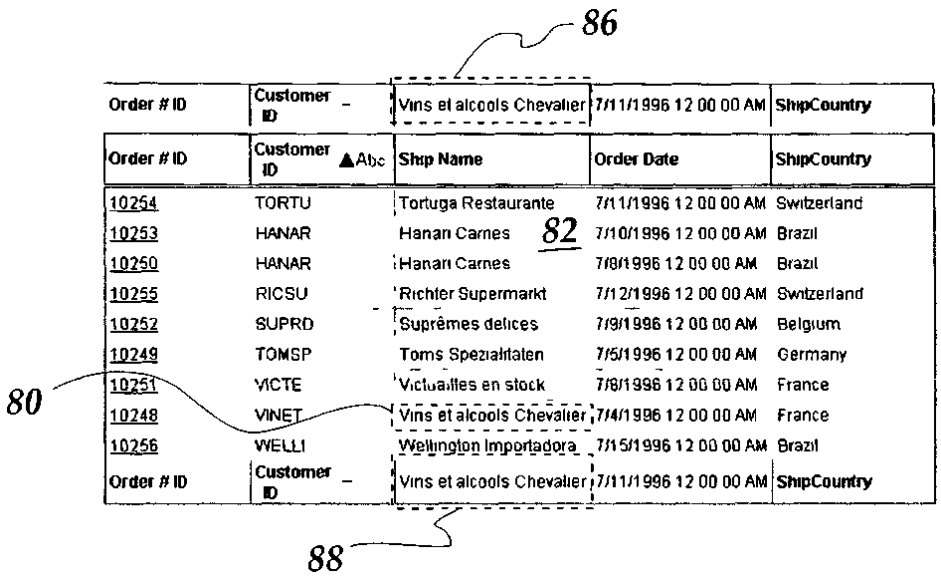
Фиг. 3



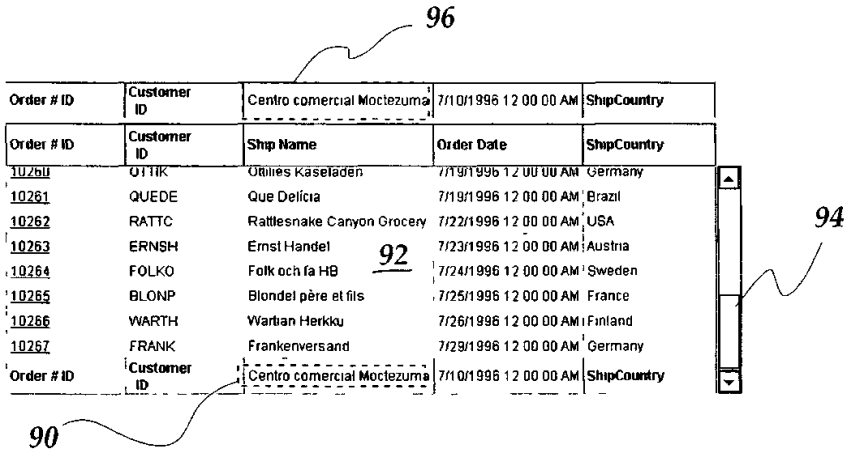
Фиг. 4



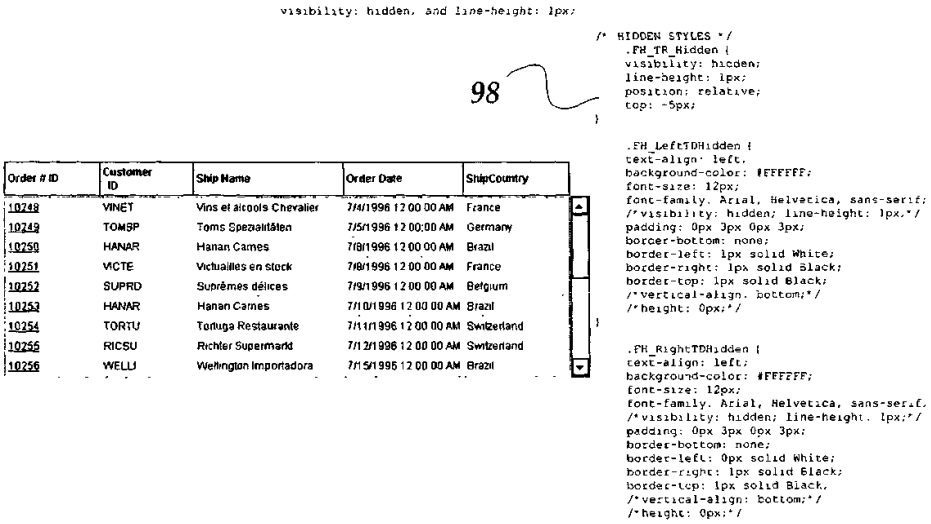
Фиг. 5



Фиг. 6

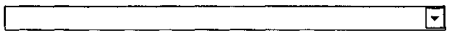


Фиг. 7

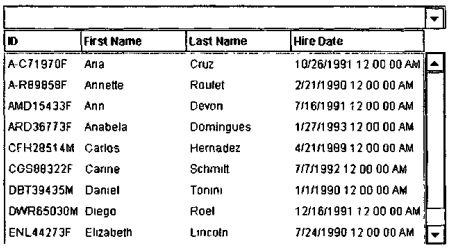


Фиг 8

Сжатое состояние

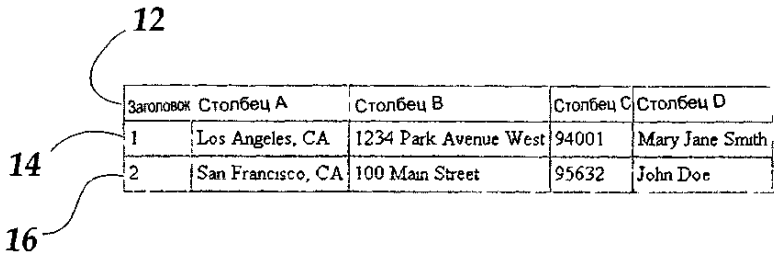


Развернутое состояние



ID	First Name	Last Name	Hire Date
A-C71970F	Ana	Cruz	10/26/1991 12 00 00 AM
A-R89858F	Annette	Roufet	2/21/1990 12 00 00 AM
AMD15433F	Ann	Devon	7/16/1991 12 00 00 AM
ARD36773F	Anabela	Domingues	1/27/1993 12 00 00 AM
CFH28514M	Carlos	Hernadez	4/21/1989 12 00 00 AM
CGS88322F	Carine	Schmitt	7/7/1992 12 00 00 AM
DBT39435M	Daniel	Tonini	1/1/1990 12 00 00 AM
DWR65030M	Diego	Roel	12/16/1991 12 00 00 AM
ENL44273F	Elizabeth	Lincoln	7/24/1990 12 00 00 AM

Фиг. 9



Заголовок	Столбец A	Столбец B	Столбец C	Столбец D
1	Los Angeles, CA	1234 Park Avenue West	94001	Mary Jane Smith
2	San Francisco, CA	100 Main Street	95632	John Doe

Фиг. 10

Заголовок	Столбец A	Столбец B	Столбец C	Столбец D
1	Los Angeles, CA	1234 Park Avenue West	94001	Mary Jane Smith
2	San Francisco, CA	100 Main Street	95632	John Doe

Фиг. 11

[ASCII код] = "фактический" размер "символа"

Фиг. 12

Ширина символа для названия шрифта:
Times, размер эм 14, жирный текст (в противоположность нормальному), единицы в пикселях

[32] = (4 pixels)
[33] = ! (8 113281 pixels)
[34] = " (9 859375 pixels)
[35] = # (10 87305 pixels)
[36] = $ (10 87305 pixels)
[37] = % (14 99219 pixels)
[38] = & (12 92383 pixels)
[39] = ' (6 941406 pixels)
[40] = ((8 113281 pixels)
[41] =) (8 113281 pixels)
[42] = * (8 810547 pixels)
[43] = + (11 21875 pixels)
[44] = , (7 433594 pixels)
[45] = - (8 113281 pixels)
[46] = . (7 433594 pixels)
[47] = / (7 433594 pixels)
[48] = 0 (10 87305 pixels)
[49] = 1 (10 87305 pixels)
[50] = 2 (10 87305 pixels)
[51] = 3 (10 87305 pixels)
[52] = 4 (10 87305 pixels)
[53] = 5 (10 87305 pixels)
[54] = 6 (10 87305 pixels)
[55] = 7 (10 87305 pixels)
[56] = 8 (10 87305 pixels)
[57] = 9 (10 87305 pixels)
[58] = : (8 113281 pixels)
[59] = ; (8 113281 pixels)
[60] = < (11 21875 pixels)
[61] = = (11 21875 pixels)
[62] = > (11 21875 pixels)
[63] = ? (11 55273 pixels)
[64] = @ (16 05273 pixels)
[65] = A (12 92383 pixels)
[66] = B (12 92383 pixels)
[67] = C (12 92383 pixels)
[68] = D (12 92383 pixels)
[69] = E (12 24414 pixels)
[70] = F (11 55273 pixels)
[71] = G (13 61523 pixels)
[72] = H (12 92383 pixels)
[73] = I (7 433594 pixels)
[74] = J (10 87305 pixels)
[75] = K (12 92383 pixels)
[76] = L (11 55273 pixels)
[77] = M (14 29492 pixels)
[78] = N (12 92383 pixels)
[79] = O (13 61523 pixels)
[80] = P (12 24414 pixels)
[81] = Q (13 61523 pixels)
[82] = R (12 92383 pixels)
[83] = S (12 24414 pixels)
[84] = T (11 55273 pixels)
[85] = U (12 92383 pixels)
[86] = V (12 24414 pixels)
[87] = W (15 06602 pixels)
[88] = X (12 24414 pixels)
[89] = Y (12 24414 pixels)
[90] = Z (11 55273 pixels)
[91] = [(8 113281 pixels)
[92] = \ (7 433594 pixels)
[93] =] (8 113281 pixels)
[94] = ^ (11 21875 pixels)
[95] = _ (10 87305 pixels)
[96] = ` (8 113281 pixels)
[97] = a (10 87305 pixels)
[98] = b (11 55273 pixels)
[99] = c (10 87305 pixels)
[100] = d (11 55273 pixels)
[101] = e (10 87305 pixels)
[102] = f (8 113281 pixels)
[103] = g (11 55273 pixels)
[104] = h (11 55273 pixels)
[105] = i (7 433594 pixels)
[106] = j (7 433594 pixels)
[107] = k (10 87305 pixels)
[108] = l (7 433594 pixels)
[109] = m (14 99219 pixels)
[110] = n (11 55273 pixels)
[111] = o (11 55273 pixels)
[112] = p (11 55273 pixels)
[113] = q (11 55273 pixels)
[114] = r (8 810547 pixels)
[115] = s (10 87305 pixels)
[116] = t (8 113281 pixels)
[117] = u (11 55273 pixels)
[118] = v (10 87305 pixels)
[119] = w (13 61523 pixels)
[120] = x (10 87305 pixels)
[121] = y (10 87305 pixels)
[122] = z (10 18164 pixels)
[123] = { (8 810547 pixels)
[124] = | (7 457031 pixels)
[125] = } (8 810547 pixels)
[126] = ~ (11 21875 pixels)

[ASCII код] = "фактический" размер "символа"

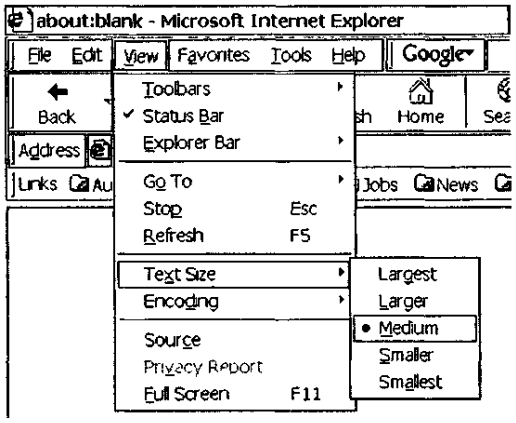
Фиг. 13

Измеренный текст New Orleans Saints
Шрифт Geneva
Размер 10 пикселей
Текст нормальный

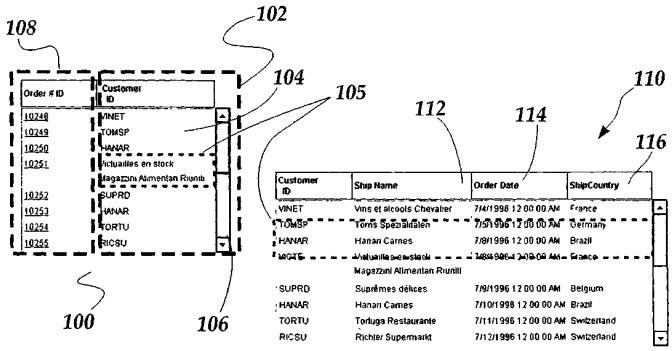
```
N    Size = 10.76986 pixels
e    Size = 9.060872 pixels
w    Size = 10.76986 pixels
     Size = 3.333333 pixels
O    Size = 11.34603 pixels
r    Size = 6.761067 pixels
l    Size = 5.681966 pixels
e    Size = 9 060872 pixels
a    Size = 9.060872 pixels
n    Size = 9.060872 pixels
s    Size = 8.4847 pixels
     Size = 3.333333 pixels
S    Size = 10.20345 pixels
a    Size = 9.060872 pixels
i    Size = 5.681966 pixels
n    Size = 9.060872 pixels
t    Size = 6.194661 pixels
s    Size = 8.4847 pixels
------------------------------
```

Общий размер = 94 21223 пикселей

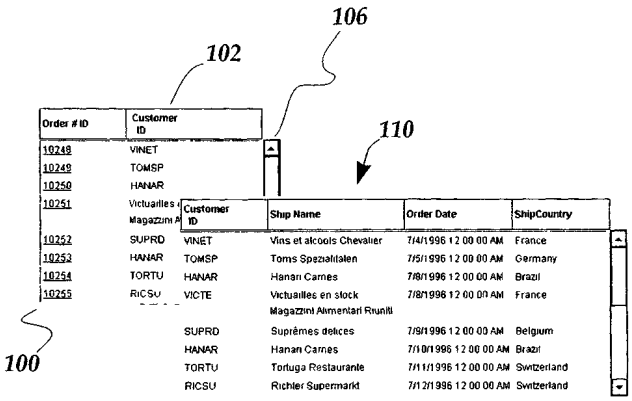
Фиг. 14



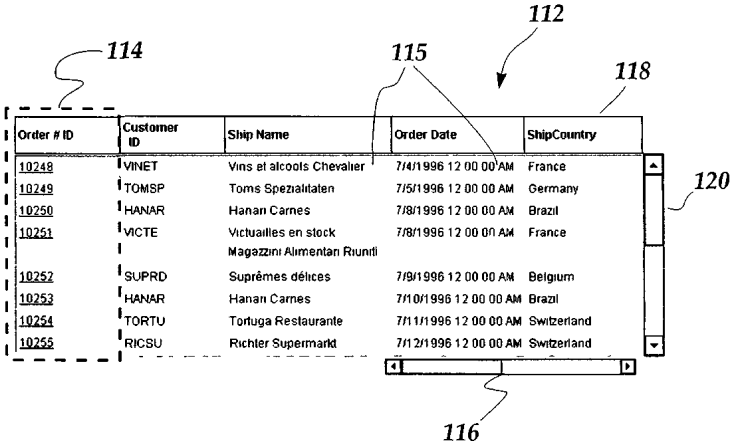
Фиг. 15



Фиг. 16-17


106
102
110
100
Order # ID
Customer ID
Ship Name
Order Date
ShipCountry

Фиг. 18


112
114
115
118
120
116
Order # ID
Customer ID
Ship Name
Order Date
ShipCountry

Фиг. 19